Date=20/07/2020
Lecture By=Shubham Joshi
Subject ⇒ Sorting

IN PREVIOUS LECTURE **(QUICK RECAP)** **Date-17/07/2020**	**In Today's Lecture (Overview)**
⇒ SLA in Software ⇒ What Is Time Complexity In Python? ⇒ What Big O Notation Means? ==>What Are The Different Big O Notation Measures?' O(1): O(log n): O(n): (n log n): O(n square): ⇒ Examples	⇒ What Is Algorithm ⇒ What is swap In Python Sorting ⇒ What is sorting in python ⇒ How to use sorting in python Types of sorting in python 1.⇒ Selection Sorting 2.⇒ Bubble sorting 3.⇒ Insertion Sort ⇒ Questions For Self Practice

⇒ What Is Algorithm

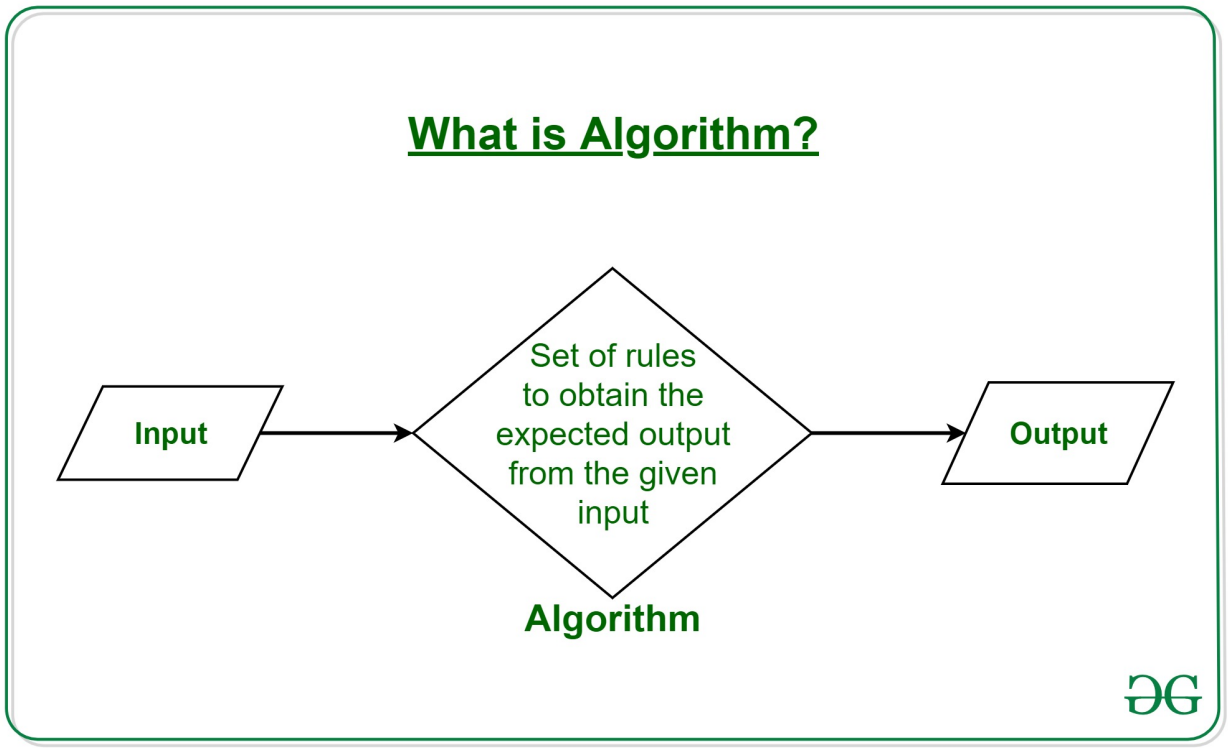
Definition=

- a process or set of rules to be **followed in calculations or other problem-solving operations,** especially by a computer.

-a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

In short

- Algorithm is **predefined** steps for software/Codes



⇒ What is swap In Python

Definition

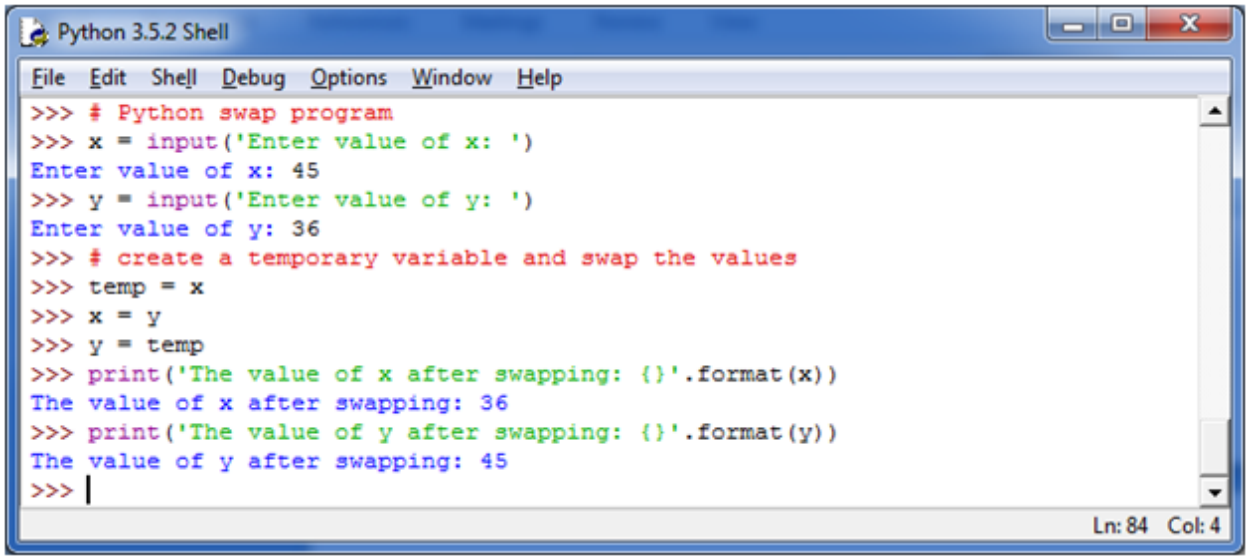
Swap means exchange **the value of Variables**

Swapping two variables refers to mutually exchanging the values of the variables. Generally, this is done with the data in memory.

Explanation;

The simplest method to swap two variables is to use a third temporary variable :

```
define swap(a, b)
   temp := a
   a := b
   b := temp
```



Sorting

⇒ What is sorting in python

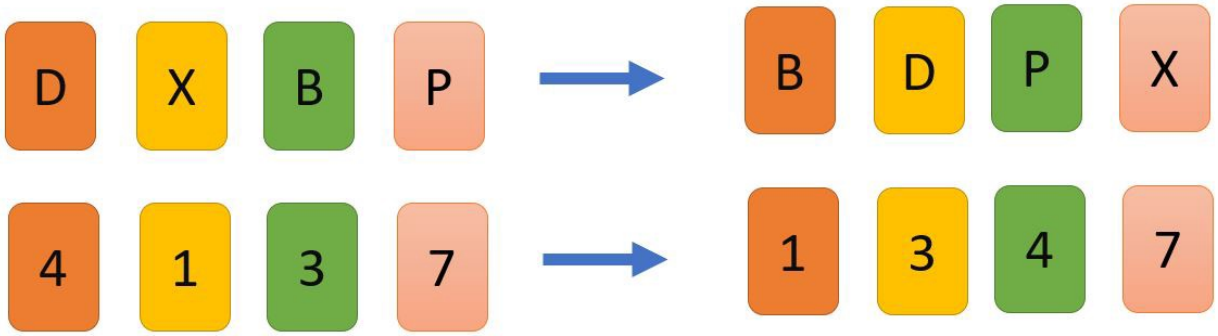
Definition

-sorting is any process that involves **arranging the data into some meaningful order** to make it easier to understand,

In short

-Sorting means to **put elements in ascending or descending order**

Sorting Algorithms



⇒ How to use sorting in python

To sort items in python you have to create list
for example;
W is list

W = [2,3,5,4,8,11,10]

For Ascending Order;
Type = sorted(W)

For descending Order;
Type= sorted(W, Reverse=true)

```
>>> l = [10, 30, 2, 5, 18]
>>> l
[10, 30, 2, 5, 18]
>>> sorted(l)
[2, 5, 10, 18, 30]
>>> sorted(l, reverse=True)
[30, 18, 10, 5, 2]
>>>
```

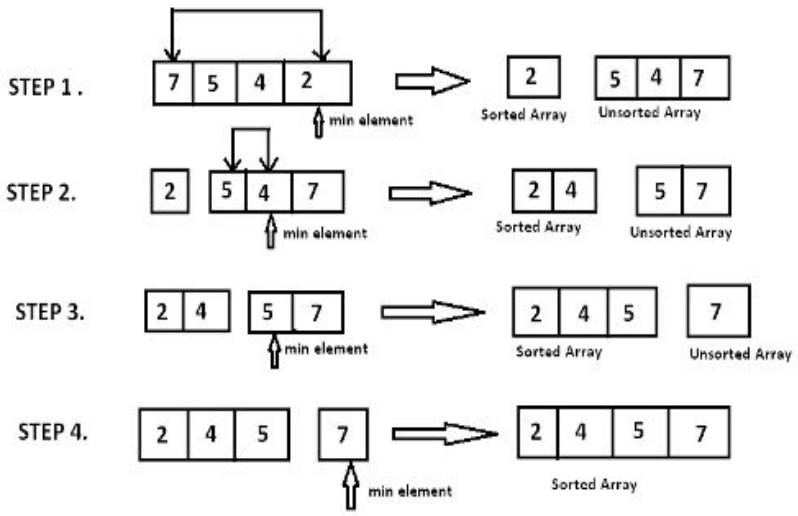
Types of sorting in python

1.⇒ Selection Sorting

Defination;
-an array is sorted by **recursively finding the minimum element from the unsorted part and inserting it at the beginning.**
-During every iteration of selection sort, the minimum element from the unsorted subarray is popped and inserted into the sorted subarray.

In short:
-this sort function **finds the minimum element and it swaps with the first element**

Selection sort in Python



https://www.geekboots.com/python/selection-sort

Practical Example From The Class

```python
a = [4,3,2,66,7,8,9]
n = len(a) # size of the list
print("list before sorting: ", a)
for i in range(0, n):
    min_ele_idx = i
    # finding the min element
    for j in range(i + 1, n):
        if a[min_ele_idx] > a[j]:
            min_ele_idx = j
    a[i], a[min_ele_idx] = a[min_ele_idx], a[i]


print("list after sorting", a)
```

"Click here" To know more About it

2.⇒ Bubble sorting

Definition;

-The idea behind Bubble Sort is very simple, **we look at pairs of adjacent elements in an array,**
-one pair at a time, **and swap their positions if the first element is larger than the second,**

Inshort;

-it picks **Up the bigger number and puts it in the last and smaller at ahead of bigger number**

For visualisation Of Bubble Sorting Click Here=**https://www.hackerearth.com/practice/algorithms/sorting/bubble-sort/visualize/**

8 5 3 1 4 7 9

Practical example From Class

```
def bubbleSort(a):
    n = len(a)
    for i in range(0, n - 1):
        j = 0
        while j < n - j - 1:
            if a[j] > a[j+1]:
                a[j], a[j+1] = a[j+1], a[j]
            j += 1
    return a


a = [2,3,4,2, 266,7,8,9]
#print(selectionSort(a))
print(bubbleSort(a))
```

"Click Here" To Know More About It

3.⇒ Insertion Sort

Explanation;

-**An array is Divided into a "sorted" subarray and an "unsorted" subarray**. At the beginning, the sorted subarray contains only the first element of our original array.

-**The first element in the unsorted array is evaluated so that we can insert it into its proper place in the sorted subarray.**

-The insertion is done **by moving all elements larger than the new element one position to the right.**

-Continue doing this until our entire array is sorted.

In short;

It takes an element from array And puts it at its Correct Position

Sorting Arrays

```
def insertion_sort(array):

    # We start from 1 since the first element is trivially sorted
    for index in range(1, len(array)):
        currentValue = array[index]
        currentPosition = index

        # As long as we haven't reached the beginning and there is an element
        # in our sorted array larger than the one we're trying to insert - move
        # that element to the right
        while currentPosition > 0 and array[currentPosition - 1] > currentValue:
            array[currentPosition] = array[currentPosition -1]
            currentPosition = currentPosition - 1


        # We have either reached the beginning of the array or we have found
        # an element of the sorted array that is smaller than the element
        # we're trying to insert at index currentPosition - 1.
        # Either way - we insert the element at currentPosition
        array[currentPosition] = currentValue
```

Let's Create a simple array and sort it:

```
array = [4, 22, 41, 40, 27, 30, 36, 16, 42, 37, 14, 39, 3, 6, 34, 9, 21, 2, 29, 47]
insertion_sort(array)
print("sorted array: " + str(array))
```

Output:

```
sorted array: [2, 3, 4, 6, 9, 14, 16, 21, 22, 27, 29, 30, 34, 36, 37, 39, 40, 41, 42, 47]
```

"Click Here"To know more about it

⇒ Questions For Self Practice

Q1. Write and practice all the 4 sorting algorithms
Bubble
Optimized Bubble
Insertion
Selection